

impara

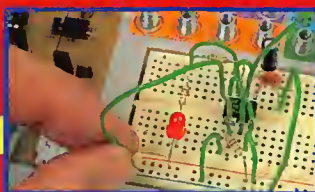
# elettronica digitale

...e costruisci il tuo **LABORATORIO DIGITALE**

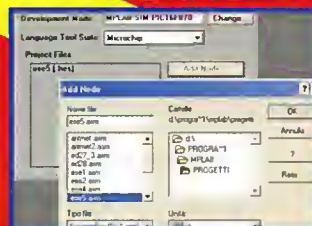
6,90 €



**HARDWARE**

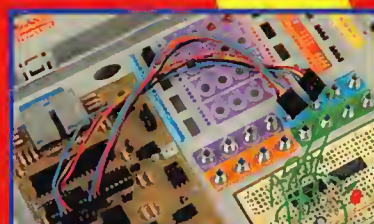


**DIGITALE DI BASE**



**MICROCONTROLLER**

37



**DIGITALE AVANZATO**



**TOTALMENTE  
PROGRAMMABILE!!!**

Peruzzo & C.





Direttore responsabile:  
ALBERTO PERUZZO  
Direttore Grandi Opere:  
GIORGIO VERCELLINI  
Consulenza tecnica  
e traduzioni:  
CONSULCOMP S.n.c.  
Pianificazione tecnica  
LEONARDO PITTON

Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (Mi). Distribuzione SO.D.I.P. S.p.A., Cinisello Balsamo (Mi).

© 2004 F&G EDITORES, S.A.  
© 2005 PERUZZO & C. s.r.l. Tutti i diritti sono riservati. Nessuna parte di questa pubblicazione può essere riprodotta, archiviata su sistema recuperabile o trasmessa, in ogni forma e con ogni mezzo, in mancanza di autorizzazione scritta della casa editrice. La casa editrice si riserva la facoltà di modificare il prezzo di copertina nel corso della pubblicazione, se costretta da mutate condizioni di mercato.

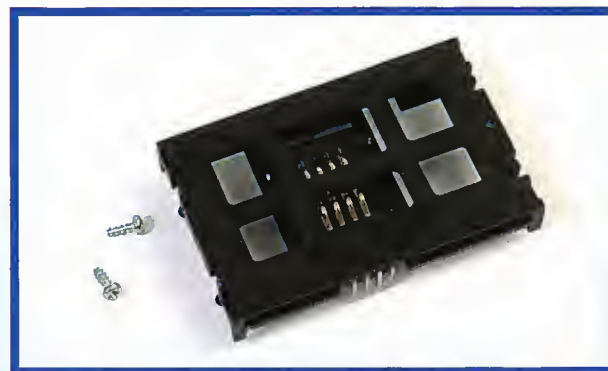
"ELETTRONICA DIGITALE"  
si compone di  
70 fascicoli settimanali  
da suddividere  
in 2 raccoglitori.

**RICHIESTA DI NUMERI ARRETRATI.** Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (Mi). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammontano a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. **IMPORTANTE:** è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

# impara l'elettronica digitale

## IN REGALO in questo fascicolo

- 1 Connettore da c.s.  
per SmartCard
- 2 Viti



## IN REGALO nel prossimo fascicolo



- 1 Scheda di memoria

## COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartelle, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede. Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail: [elettronica digitale@microrobots.it](mailto:elettronica digitale@microrobots.it)

**Hardware** Montaggio e prove del laboratorio

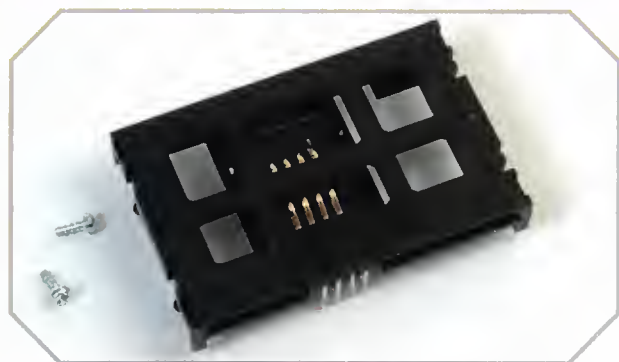
**Digitale di base** Esercizi con i circuiti digitali

**Digitale avanzato** Esercizi con i circuiti sequenziali

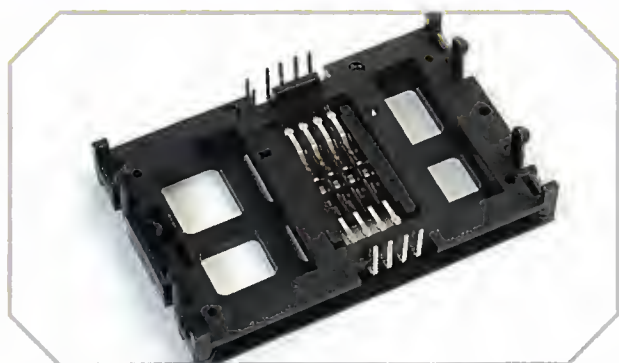
**Microcontroller** Esercizi con i microcontroller



## Lettura e scrittura di schede (II)



Componenti forniti con questo fascicolo.



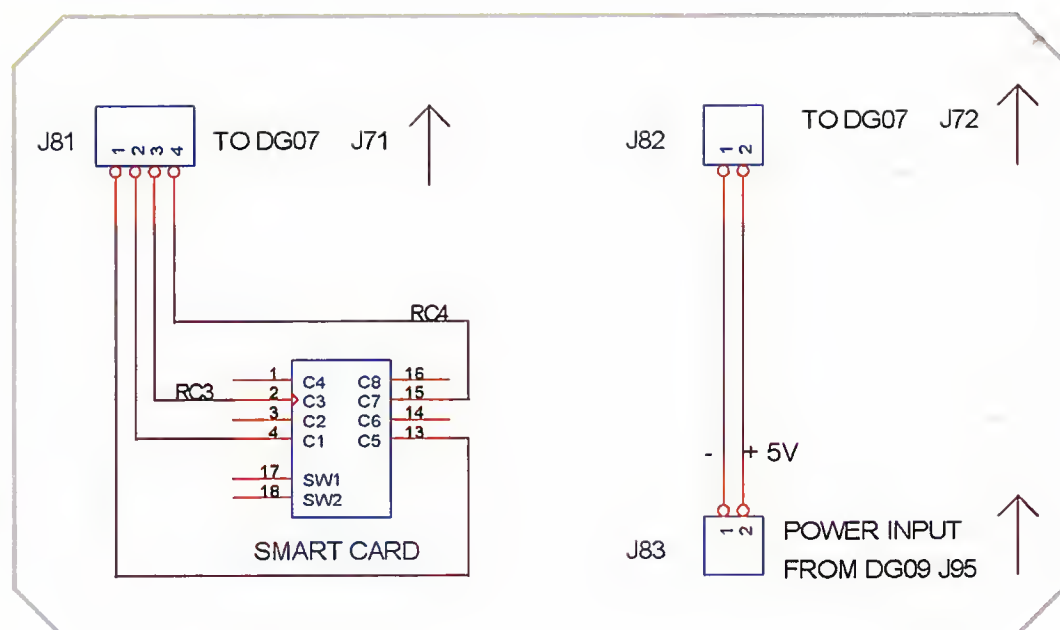
Per facilitare il montaggio lo zoccolo dispone di incastri.

**C**on questo fascicolo viene fornito il dispositivo di lettura-scrittura per schede di memoria e le due viti per installare il circuito stampato DG08. Terminiamo il montaggio della scheda DG08, poi la installeremo in modo definitivo sul laboratorio.

Con il montaggio di questa terza scheda, dato che le DG06 e DG07 sono già montate, si completa la zona 2 del laboratorio digitale. Questa zona è destinata a supportare gli esperimenti nei quali è necessario lavorare con un programma scritto sul PIC 16F870, oppure eseguire la scrittura del programma sul PIC. Può anche essere utilizzata per la scrittura della SmartCard o scheda di memoria e per trasferire informazioni tra la SmartCard e il PIC, però tutte queste funzioni vi verranno spiegate dettagliatamente in ogni esperimento per facilitare la realizzazione degli stessi.

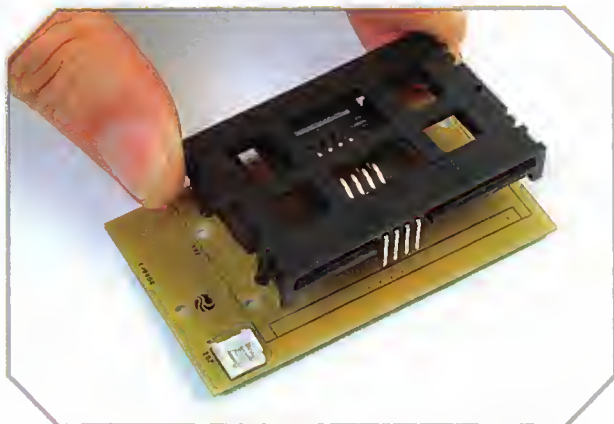
### Centratura dello zoccolo

Il montaggio di questo circuito stampato termina con l'installazione del dispositivo letto-

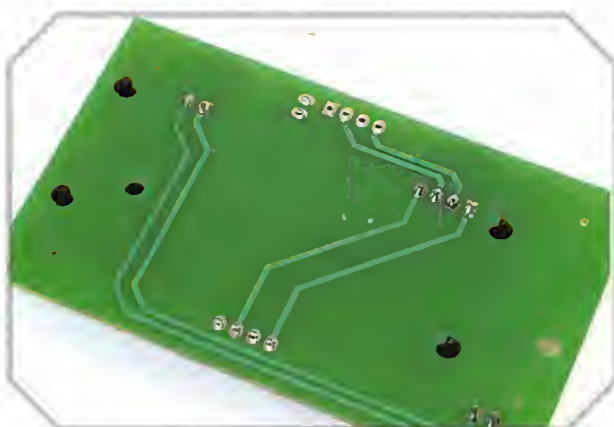


Schema elettrico della scheda DG08.

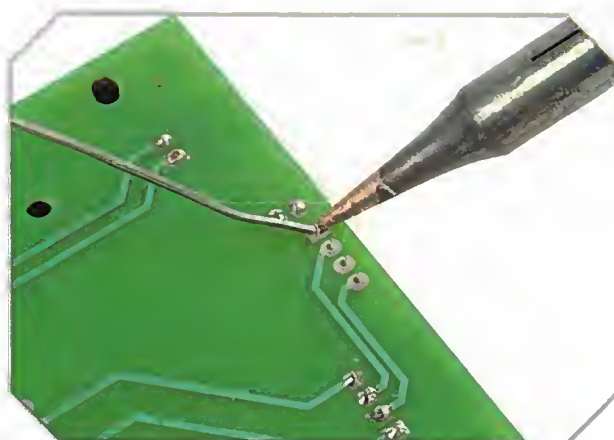




*Il circuito stampato è progettato appositamente per questo zoccolo.*



*Incastri dello zoccolo visti dal lato delle saldature.*



*Saldatura dei terminali dello zoccolo.*

re-scrittore di schede, che chiameremo in modo abbreviato zoccolo; questo dispositivo si può montare solamente in una posizione pre-determinata sul circuito stampato.

Prima di montare questo dispositivo è necessario verificare che i terminali di collegamento dello stesso siano ben diritti, in caso contrario vanno raddrizzati con attenzione perché non sopportano molte flessioni, utilizzeremo allo scopo un paio di piccole pinzette a punta piatta.

Dopo aver allineato correttamente i terminali avvicineremo lo zoccolo al circuito stampato, assicurandoci attentamente che ogni terminale del connettore entri con facilità nel foro corrispondente.

Questo zoccolo dispone, ai suoi lati, di alcune guide che fanno parte dello zoccolo stesso, servono per la centratura e il fissaggio e devono essere inserite nei fori predisposti sul circuito stampato.

Dopo aver verificato che tutti gli elementi siano correttamente allineati, inseriremo con attenzione lo zoccolo sul circuito stampato fino a inserire gli incastri nelle loro sedi e fissare così lo zoccolo; verificheremo nuovamente che tutti i terminali del connettore fuoriescano dal lato delle saldature del circuito stampato.

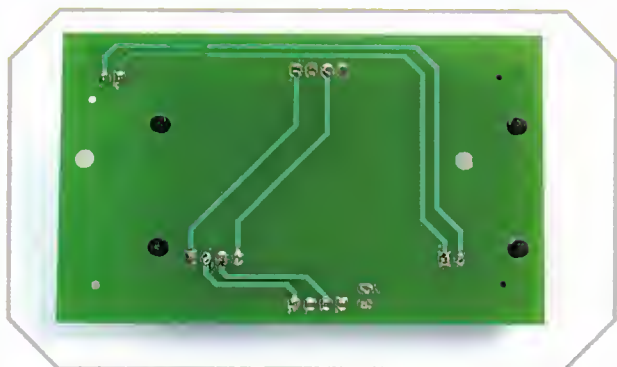
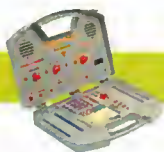
## Saldatura

Dopo aver verificato il corretto posizionamento dello zoccolo procederemo a saldarne i terminali nel modo abituale, applicando il saldatore per il tempo sufficiente a ottenere una buona saldatura. Non bisogna applicare per troppo tempo il saldatore, in quanto lo zoccolo si potrebbe deformare rendendo difficile in seguito il collegamento con i terminali piatti di contatto della SmartCard.

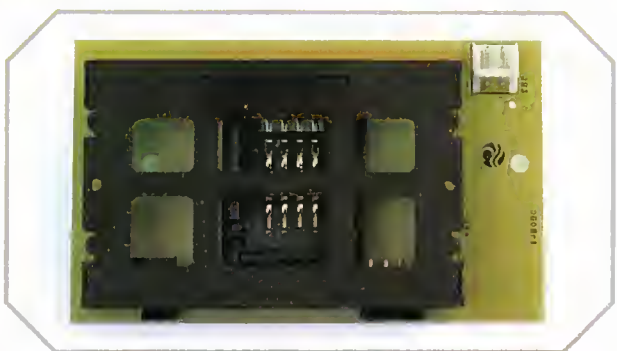
Dopo aver terminato verificheremo tutte le saldature, comprese quelle realizzate in precedenza, controllando che siano state eseguite correttamente, senza averne tralasciata nessuna e che non esistano cortocircuiti fra punti o zone di saldature vicine fra loro.

## Collegamento a DG07

Il collegamento alla scheda DG07 si esegue tramite i connettori J81 e J82, che si collegano loro volta con i corrispondenti J71 e J72 della



Vista generale del lato saldature.



Scheda DG08 terminata.

scheda DG07. La coppia J72 e J82 supporta il collegamento dell'alimentazione, mentre J71 e J81 il collegamento dello zoccolo del dispositivo di lettura-scrittura della SmartCard.

Il collegamento fra le due schede si esegue allineando bene i connettori situati su entrambe le schede e inserendo la scheda DG08 sino a collegamento avvenuto.

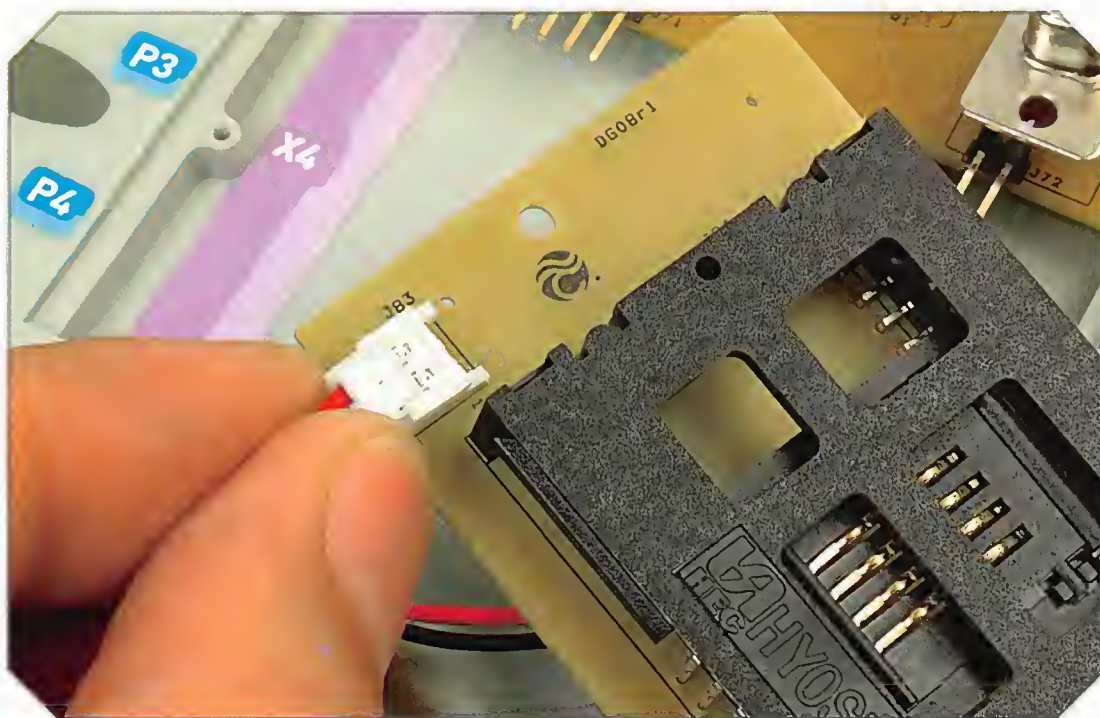
### Collegamento dell'alimentazione

Il connettore di ingresso dell'alimentazione è J83, di colore bianco, situato sul lato opposto della scheda. Il terminale 1 corrisponde al negativo, e il 2 al positivo.

In questo connettore bisogna collegare il cavetto a due terminali che arriva dalla parte inferiore del pannello principale del laboratorio.

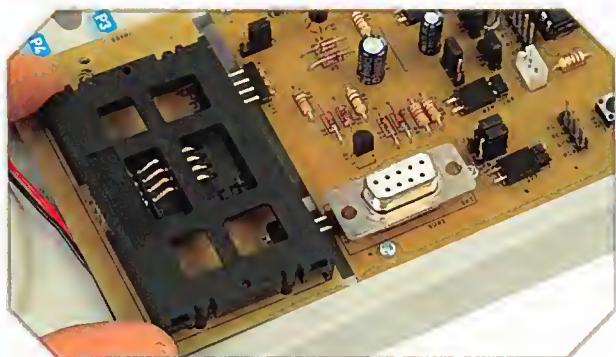
Il connettore del cavetto si deve inserire con forza sino a quando si collega e rimane perfettamente fissato a quello della scheda, il filo nero corrisponde al negativo e quello rosso al positivo.

Questo cavetto prende l'alimentazione da 5 a 4,5 V quando è alimentato a batterie, dalla scheda di distribuzione dell'alimentazione DG09.

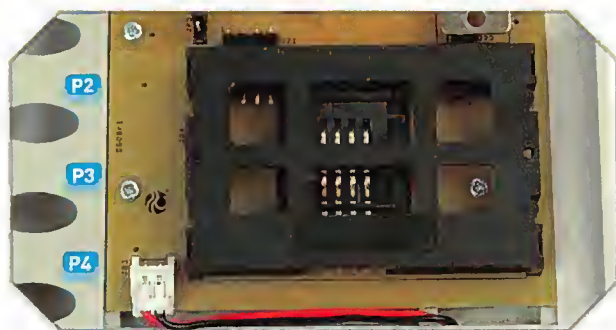


Collegamento del cavetto di alimentazione.





Collegamento tra DG07 e DG08.



Viti di fissaggio della scheda.

## Montaggio della scheda

Dopo aver collegato la scheda DG08 possiamo vedere che i due fori di fissaggio di questa scheda, sono allineati con le torrette forate, situate sul pannello principale del laboratorio, e identificate con le sigle X4 e X8, destinate a ospitare le viti.

Una delle due viti si trova in una zona accessibile e comoda, ed è di facile installazione: la dobbiamo inserire e ruotare di circa due giri, in modo che mantenga una certa mobilità per facilitare l'inserimento dell'altra vite, che è posizionata all'interno dello zoccolo e richiede un po' di abilità in più per fissarla, centrando il foro con la colonna; potrebbe essere utile aiutarci con delle pinzette per posizionare la seconda vite.

Dopo aver inserito le due viti le avviteremo fino a fissare la scheda, senza stringere troppo per non danneggiare il filetto che ogni vite si crea nella plastica.

Se non utilizziamo questa zona dobbiamo togliere l'alimentazione alla stessa, e il modo più semplice per farlo è quello di togliere i ponticelli della scheda DG06 e non utilizzare il collegamento al PC.

Laboratorio completato.





# Monostabile con 555

**Q**uesto circuito corrisponde a una delle applicazioni più utilizzate del 555, è basato su un circuito applicativo consigliato dal costruttore.

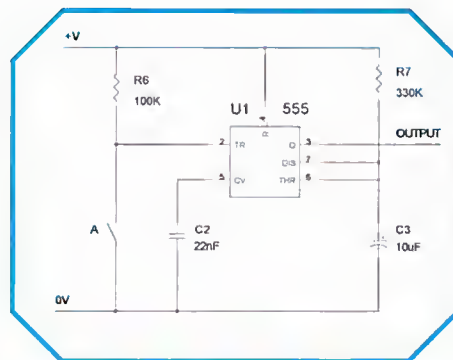
## Il circuito

Se guardiamo lo schema del circuito possiamo notare due parti molto importanti: il circuito di attivazione e quello di temporizzazione.

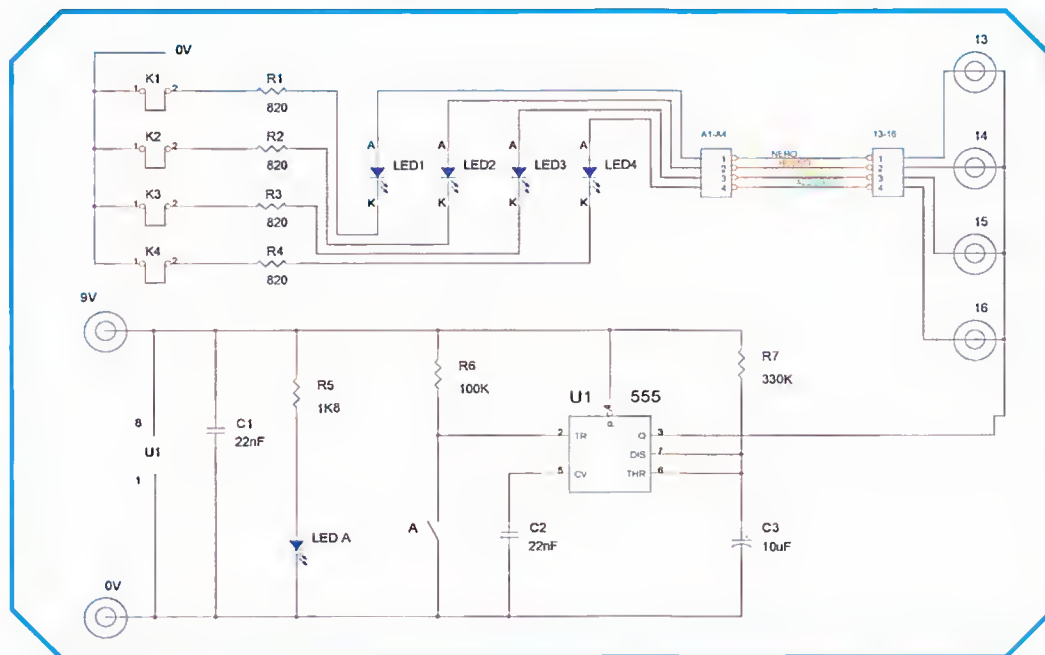
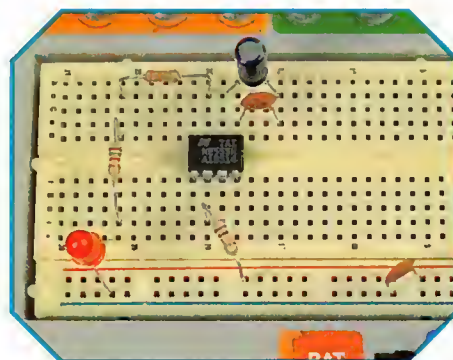
Nello stato di riposo l'uscita del circuito rimane a livello basso e, una volta effettuata l'attivazione, l'uscita passa a livello alto, rimanendo in questo stato il tempo determinato dai componenti del circuito, trascorso il quale passa a livello basso e permane in questo stato fino a quando non avviene una nuova attivazione.

Il circuito di attivazione è composto da una resistenza, R6, che mantiene il terminale di attivazione – terminale 2 dell'integrato – a livello alto, per evitare attivazioni impreviste. L'attivazione avviene quando questo terminale dell'integrato passa per un attimo a livello basso, normalmente si tratta di un pulsante, ma potrebbe essere anche un segnale di livello basso generato da un altro circuito. Il nostro circuito sperimentale non dispone, per il momento, di un pulsante, utilizzeremo quindi un collegamento con un filo indicato nello schema dalla lettera A.

Schema di base del monostabile.

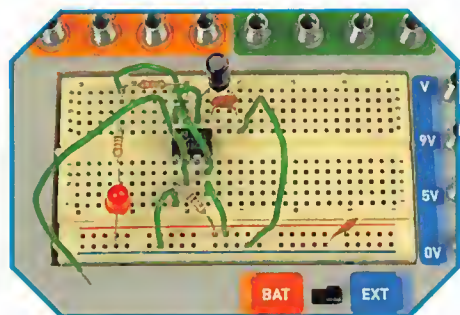
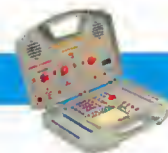


Il 555 e alcuni componenti montati sulla scheda.

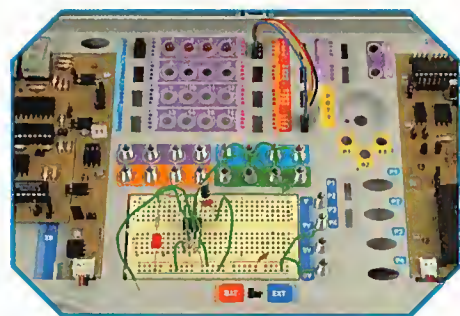


Monostabile con 555.

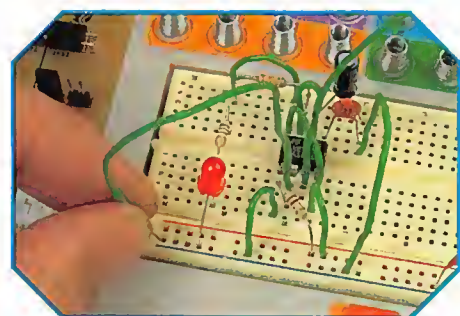




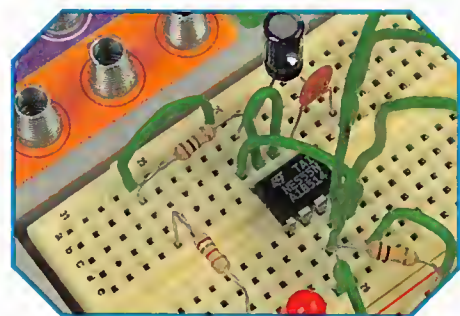
*Iniziamo il cablaggio sulla scheda Bread Board.*



*Vista del cablaggio completato.*



*Chiudendo per un istante il collegamento A si attiva il monostabile.*



*Aumentando R7 a 1 M si incrementa la durata dell'impulso.*



*Esperimento completato.*

La temporizzazione è quindi, il tempo in cui il segnale di uscita – terminale 3 dell'integrato – rimane a livello alto, questo tempo è determinato dal valore della resistenza R7 e dalla capacità del condensatore C3.

$$T = 1,1 \times R7 \times C3$$

Bisogna tener presente che questa formula è approssimata, specialmente nel caso di utilizzo di condensatori elettrolitici dove la tolleranza accettata è generalmente molto ampia, a volte va da -20% a +50%, quindi bisogna verificare in pratica se la durata dell'impulso è quella adeguata alla nostra esigenza o se, al contrario, è necessario cambiare il valore di qualche componente, normalmente la resistenza R, per ottenere il valore desiderato.

Lo stato dell'uscita del circuito, terminale 3, si verifica mediante l'accensione dei LED, che sono collegati all'uscita stessa.

Il condensatore C2, consigliato dal costruttore, evita le variazioni sull'ingresso di modulazione del circuito, rendendo più stabile il funzionamento dello stesso.

Il LED A è utilizzato per verificare la presenza dell'alimentazione, dato che il circuito deve rimanere inattivo fino a quando non si agisce sul terminale di attivazione A.

## Montaggio

Questo montaggio si esegue seguendo lo schema. Non dimentichiamo che i terminali di alimentazione dell'integrato 555 sono: 1 per il negativo, che si collega a 0 V; 8 per il positivo che si collega a 5, 9 o 12 V; infine il terminale di reset, terminale 4, deve essere collegato a livello alto in modo da inibirne il funzionamento. Si utilizza un cavetto a quattro fili tra gli anodi dei 4 LED e i collegamenti dal 13 al 16 delle mole, e anche i quattro ponticelli sui catodi dei LED.

## Prova

Il circuito deve funzionare non appena si collega l'alimentazione. Quando ciò avviene il circuito rimane inattivo, si illuminerà solamente il LED che utilizziamo per verificare la reale presenza dell'alimentazione. Eseguendo per un attimo il collegamento A, i LED si illuminano e rimangono illuminati il tempo determinato dai valori scelti per la resistenza R7 e per il condensatore C3.





# Preselezione digitale

*In molti casi è necessario che il conteggio non arrivi al numero più alto che può gestire il contatore. Per questo, è necessario rilevare il numero successivo a quello che vogliamo far contare al contatore. Questa procedura è nota come preselezione digitale.*

## Il circuito

Mediante questo circuito vogliamo realizzare un esperimento con il contatore a due digit montato nella zona 1 del laboratorio. Costruiremo un contatore capace di eseguire un conteggio ascendente fra 00 e 04.

Utilizzeremo a questo scopo, diverse porte NAND per rilevare il numero cinque sul contatore delle unità.

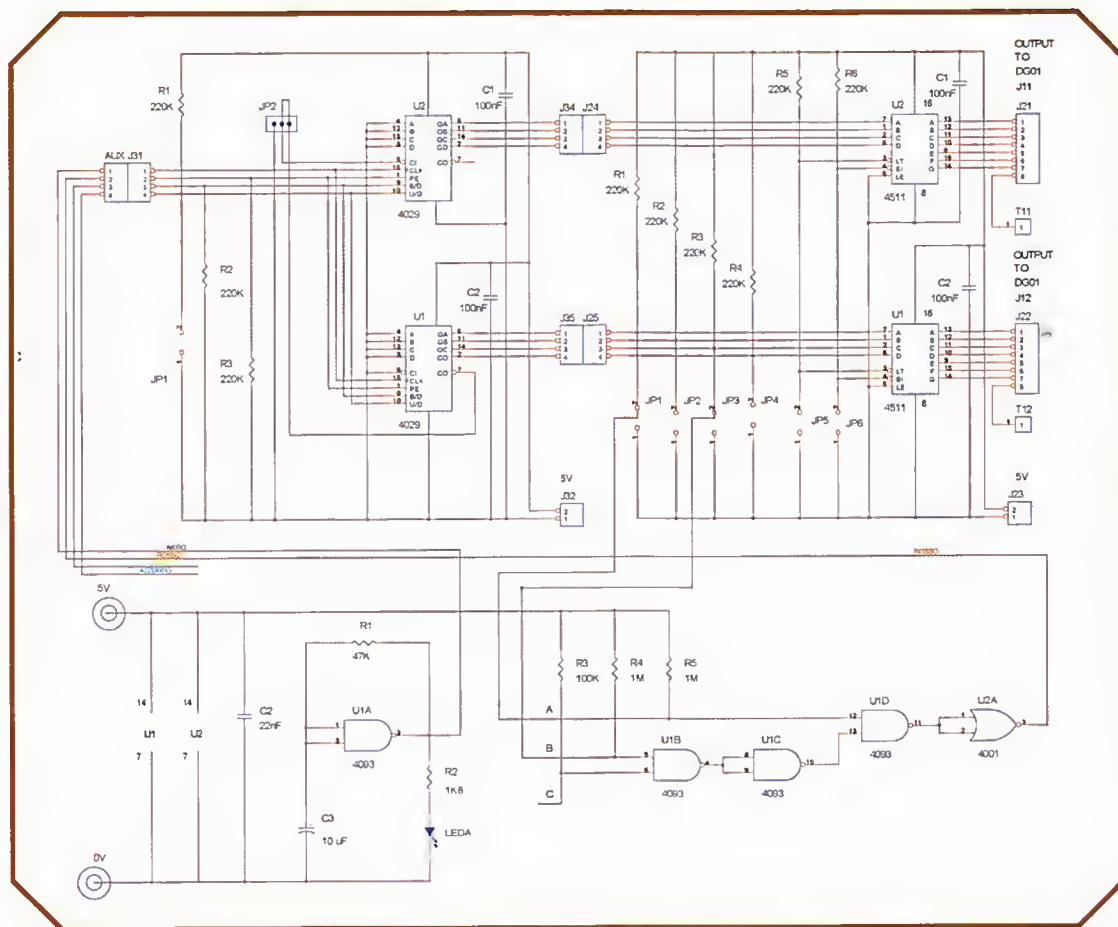
In questo caso, dei due contatori (decine e unità) rileveremo solamente un numero sul contatore delle unità che conterà in modo ascendente. Come massimo quindi, potremo fare un conteggio fra 01 e 08, e come abbia-

mo già detto, in questi due casi dovremo rilevare 02 e 09.

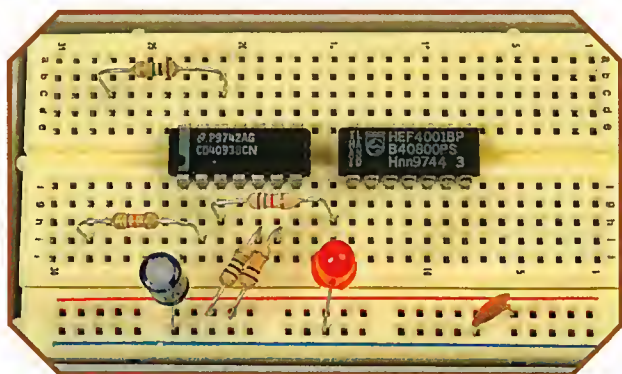
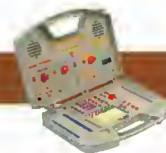
Il riconoscimento del numero successivo al conteggio, verrà fatto mediante le quattro porte NAND. In questo modo è possibile resettare il contatore e farlo iniziare nuovamente da zero.

## Clock

Il generatore di impulsi o generatore di segnale di clock, è una configurazione già nota realizzata con una porta NAND del circuito integrato 4093, una resistenza R1 da 47 K e un condensatore C3 da 10  $\mu$ F. L'uscita è presa sul-



Schema del montaggio completo.



Componenti sulla scheda Bread Board.

l'uscita della porta U1A, che corrisponde al terminale 3 del circuito integrato U1. Lo stato dell'uscita (1 o 0) verrà visualizzato sul diodo LEDA. La resistenza R2 limita la corrente che circola sul LED. Il condensatore C2 filtra l'alimentazione.

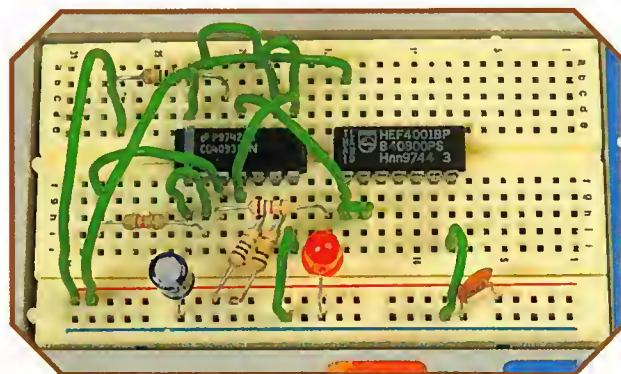
## Preselezione digitale

Non c'è nulla di meglio per capire il funzionamento del circuito che vedere un esempio pratico. Facciamo una preselezione digitale del numero cinque, in modo che il conteggio del contatore sia da 00 a 04.

Per il riconoscimento del numero cinque, rileviamo i due bit A e C che diventano '1' quando appare il numero cinque,  $5 = 0101$ .

Il circuito che abbiamo realizzato è una porta AND a tre ingressi a partire da tre porte AND a due ingressi e una porta NOR che funziona come porta invertente.

Nel caso per il quale abbiamo progettato il



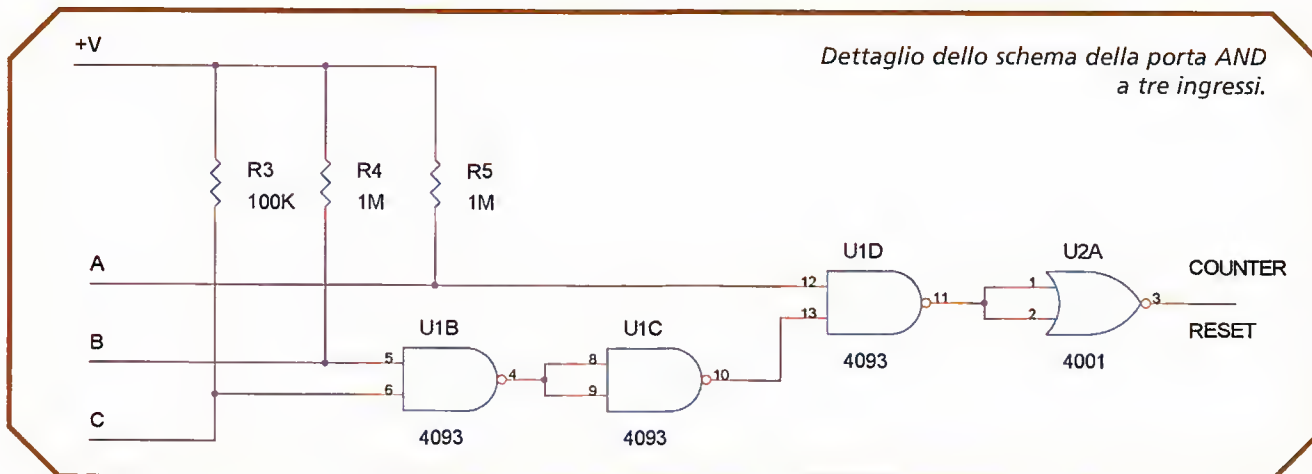
Cablaggio interno della scheda.

circuito, quando appare il numero cinque sul contatore, due dei tre ingressi della porta AND a tre ingressi – quelli segnati come A e B – avranno un livello alto '1' e il terzo lo avrà perché dispone di una resistenza di pull-up, indicata come R3.

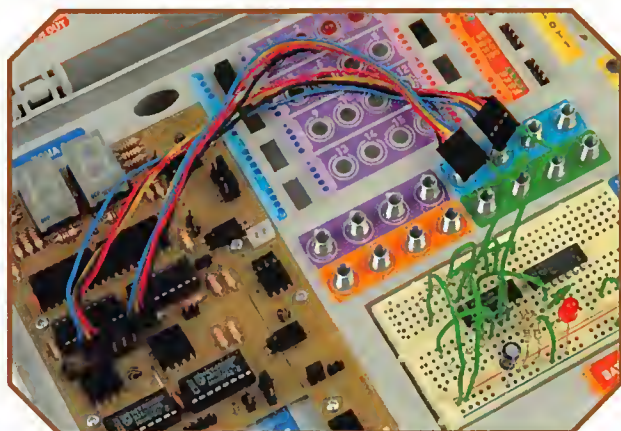
## Il circuito

Se osserviamo lo schema del circuito vedremo sulla parte sinistra l'oscillatore che genera la frequenza di clock. Questo oscillatore è realizzato con una porta NAND del tipo Trigger Schmitt (circuito integrato 4093), perché con una porta NAND convenzionale non oscilla.

La porta AND a tre ingressi è costruita nel seguente modo: con le porte U1B e U1C si fa una porta AND a due ingressi (NAND seguito da una porta invertente). L'uscita di U1C si applica a uno degli ingressi di U1D. L'altro ingresso di questa porta sarà il terzo della porta AND. L'uscita si deve invertire, quindi utilizza-







Collegamento dei fili su JP1 e JP3. Questo collegamento va effettuato in modo particolare, utilizzando solamente un filo per ognuno dei due cavetti da 4 fili, collegati su JP1 e JP3.

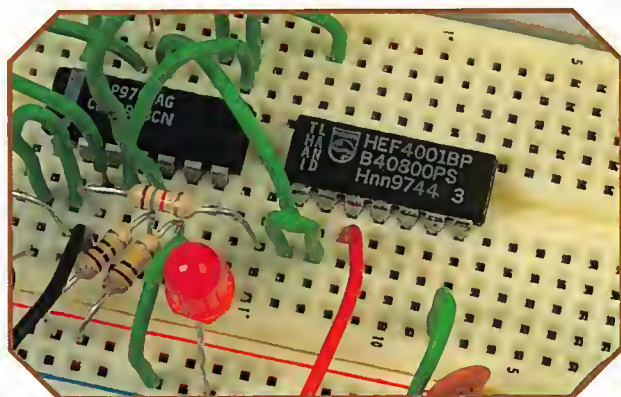
remo una porta NOR (U2A) con i due ingressi uniti.

L'uscita si applica all'ingresso PE del contatore per fare in modo che carichi il valore 0000 sulle uscite, in altre parole si esegue un reset del contatore per impostarlo a zero.

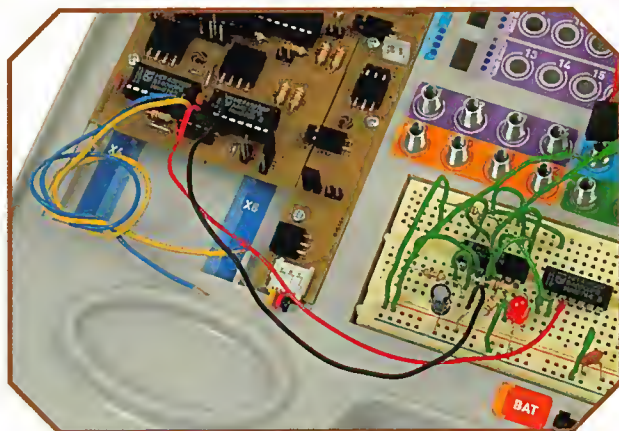
## Montaggio

Prima di iniziare il montaggio dobbiamo verificare che l'alimentazione sia scollegata, per questo vi raccomandiamo di lasciare per ultimo il collegamento al positivo dell'alimentazione della scheda Bread Board.

Dato che il contatore è già disponibile, è sufficiente montare i componenti del generatore di impulsi sulla scheda Bread Board ed eseguire il cablaggio, facendo particolare attenzione al montaggio dei circuiti integrati,



Collegamento del filo rosso di reset.

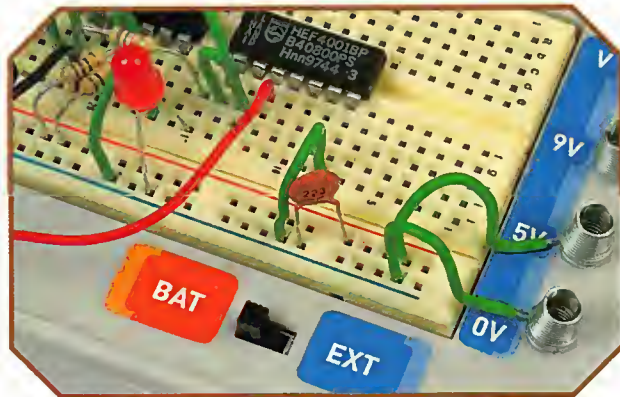


Collegamento del cavetto fra J31 di DG03 e la scheda Bread Board.

alla polarità dei condensatori elettrolitici, e a quella del diodo LED.

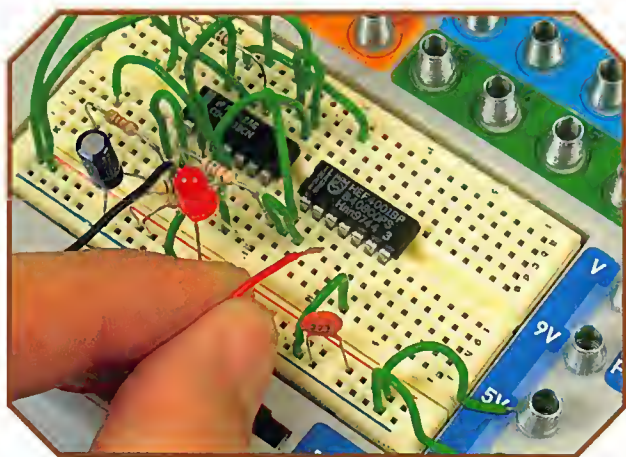
Il collegamento al contatore si realizza con un cavetto a quattro fili, terminato su di un connettore nero a quattro vie, che si collega direttamente al connettore J31 della scheda DG03, facendo attenzione a collegare il filo nero al terminale 1. Il filo rosso si deve collegare all'uscita della porta NOR. Non deve essere collegato nessun ponticello sulle schede DG01 e DG02, invece sulla DG03 bisogna inserire JP2 in posizione normale, potete verificare questa posizione dalle fotografie.

Bisogna anche tener presente i collegamenti fra i terminali 5 e 12 del circuito integrato 4093 sino ai terminali di JP1 e JP3. Questi collegamenti si possono eseguire solamente nel modo che si può osservare nelle fotografie,



Collegamento dell'alimentazione a 5 V e commutatore su BAT.





Con l'interruzione del filo rosso, il conteggio non si ferma.

anche se si utilizza solamente uno dei quattro fili per ogni cavetto.

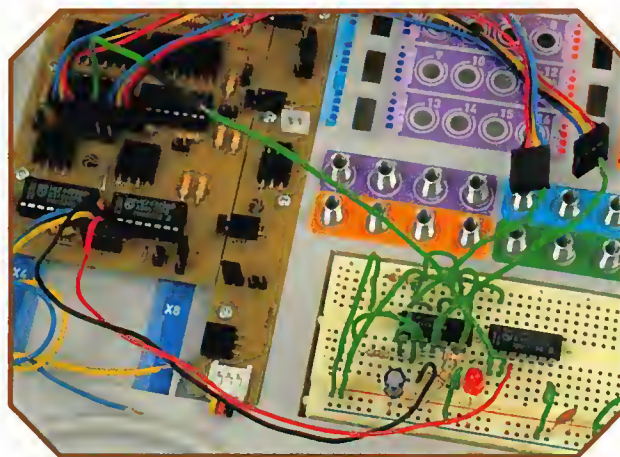
## Alimentazione

Sia il generatore di impulsi che il contatore, possono funzionare a 5 o a 9 V, però per fare in modo che i livelli di uscita dei generatori di impulsi siano compatibili con gli ingressi del contatore, entrambi i circuiti devono essere alimentati alla stessa tensione.

La tensione di alimentazione del contatore si seleziona con i ponticelli delle schede DG04 e DG05.

Per 5 V i due ponticelli della scheda DG04 devono essere in posizione 1-2, mentre sulla DG05 il ponticello deve essere montato su JP1.

Per alimentare a 9 V i ponticelli della scheda DG04 devono essere nella posizione 3-4, men-



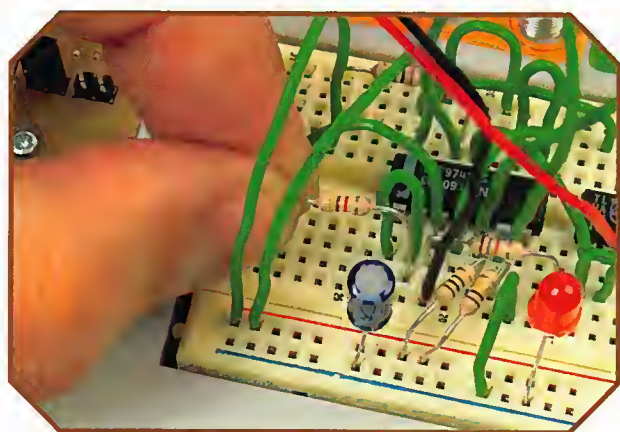
Perchè il conteggio avvenga fra 00 e 06 collegate i 3 ingressi A, B e C della porta AND su JP1, JP2 e JP3.

tre sulla DG05 il ponticello deve essere su JP2.

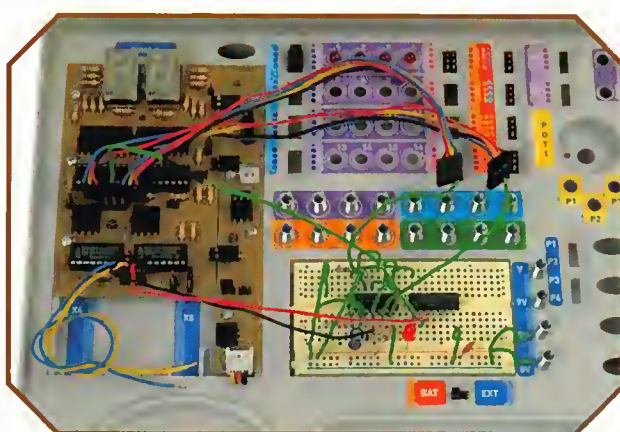
Nel caso dell'oscillatore si utilizza la molla di alimentazione da 5 V o quella da 9 V, senza dimenticare di collegare il negativo dell'alimentazione a 0 V.

### LISTA DEI COMPONENTI

U1	Circuito integrato 4093
U2	Circuito integrato 4001
R1	Resistenza 47 K (giallo, viola, arancio)
R2	Resistenza da 1K8 (marrone, grigio, rosso)
R3	Resistenza 100 K (marrone, nero, giallo)
R4, R5	Resistenza 1 M (marrone, nero, verde)
C1	Condensatore 22 nF
C2	Condensatore 10 µF elettrolitico
LEDA	Diodo LED rosso 5 mm



Abbassando il valore di R1 il contatore conta più velocemente.



Vista del laboratorio con l'esperimento completato.





## Esercizio 5: display, il programma

**A**ffrontiamo in questo caso un nuovo esercizio in cui vogliamo lavorare con il display. Prima di continuare a studiare i dispositivi del microcontroller, è necessario fare pratica e affinare le conoscenze acquisite fino a questo momento. Le difficoltà andranno crescendo man mano che vedremo i nuovi dispositivi ed è per questa ragione che dobbiamo acquisire scioltezza nel risolvere i progetti.

### Enunciato

Vogliamo visualizzare lo stato di un interruttore sul display a 7 segmenti del laboratorio. Quindi se l'interruttore è a 5 Vdc (positivo) sul display si visualizzerà "1" e se è a 0 Vdc (negativo) sul display visualizzeremo uno "0". Potremo inoltre accendere il punto decimale del display mediante un altro interruttore.

### Organigramma

L'organigramma generale dell'applicazione è molto semplice, perché dispone di una sola condizione per ogni ingresso. In questo modo, leggendo il primo ingresso, lo stato che vogliamo rappresentare, in funzione del valore letto, selezionerà una o l'altra uscita. Fatto questo, leggeremo l'ingresso successivo, per accendere nel caso sia necessario, il punto decimale.

Nella figura potete vedere l'organigramma che riassume quanto spiegato.

### Codice

Dato che abbiamo bisogno di due ingressi, utilizzeremo come fatto sinora, i terminali della porta C, RC0 e RC1 e le uscite le manderemo sulla porta B. In questo modo abbiamo definito i dispositivi del microcontroller che vogliamo utilizzare. Se iniziamo a creare il nostro codice definiremo il processore e la libreria dei dispositivi dopodiché continueremo con la definizio-

```
;Il display a 7 segmenti
;sul display a catodo comune collegato alla porta B, visualizziamo lo stato
;logico "0" o "1" dell'interruttore RC0. Mediante l'interruttore RC1 si attiva o meno il
;punto decimale.

List      p16f870      ;Rigo de processore
include "P16F870.INC" ;definizione dei registri interni

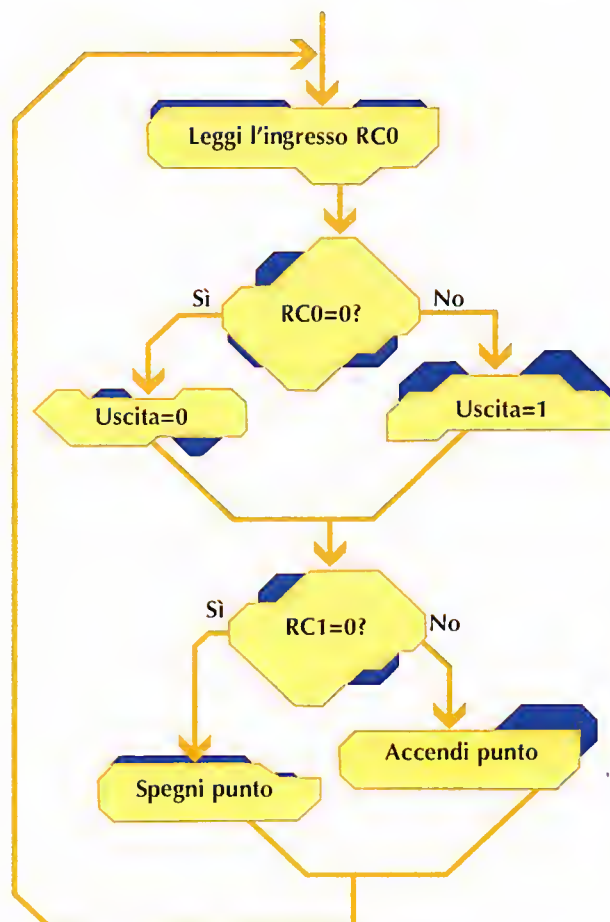
org 0x00
goto inizio
org 0x05

inizio    cllrf PORTB      ;Cancella i valori casuali
          bsf STATUS,RP0   ;Seleziona il banco 1
          cllrf TRISB      ;Porta B si configura come uscita
          movlw 0xFF        ;Porta B si configura come uscita
          movwf TRISC       ;Porta C si configura come ingresso
          bcf STATUS,RP0    ;Seleziona il banco 0
```

Codice risultante dopo aver definito i dispositivi.

ne della porta C come ingresso e della porta B come uscita. Se a tutto questo aggiungiamo i commenti pertinenti, otterremo il codice mostrato nella figura in basso.

A partire da questo punto dobbiamo definire il codice che risolve l'enunciato. Dato che abbiamo solamente due stati di ingresso, possiamo assegnarli a una determinata uscita. Nel caso di ingressi con più stati (più ingressi) potremmo fare una subroutine in cui includere una tabella di conversione ingresso/uscita, ma non essendo il caso, assegneremo diretta-



Organigramma generale dell'applicazione.



mente un'uscita ad ogni ingresso.

Nella tabella della figura a lato ricordiamo la conversione fra binario e codice a 7 segmenti in modo da assegnare l'uscita corretta.

Leggeremo il primo ingresso (RC0) utilizzando un salto condizionale, quindi se nell'istruzione successiva inseriremo goto o salto incondizionale, avremo contemplato i due possibili stati. A partire dall'indirizzo indicato da goto, risolveremo uno degli stati, però se non si esegue il goto, nell'indirizzo successivo a esso, risolveremo l'altro stato.

Al termine di ognuno di questi percorsi dovremo saltare all'altra parte del codice, dove elaboreremo l'altro ingresso, che verrà trattato nello stesso modo del primo. Fatto questo dovremo tornare all'inizio, e leggere nuovamente il primo ingresso, per costruire un programma ciclico.

Il codice finale che presentiamo è quello

	DIGIT	BINARIO	7 SEGMENTI b'0gfedcba
	0	0000	00111111
	1	0001	00000110
	2	0010	01011011
	3	0011	01001111
	4	0100	01100110
	5	0101	01101101
	6	0110	01111101
	7	0111	00000111
	8	1000	01111111
	9	1001	01100111
	punto		10000000

Conversione da binario in codice a 7 segmenti.

mostrato nella figura in basso, vi consigliamo però di non copiarlo subito, ma di risolvere da soli questo esercizio.

Sul CD potrete trovare questo stesso programma, preparato però per lavorare con il Bootloader.

Salvate il programma con estensione ".asm" nella directory di lavoro di MPLAB, per procedere alla sua compilazione e all'assemblaggio.

## Compilazione

Continuiamo con la stessa dinamica degli esercizi precedenti, con l'unico scopo di abituarci a questo modo lavoro, quindi procederemo alla compilazione. Aprite MPLAB e create un progetto. Ricordate di dare lo stesso nome al progetto e al file assembler, per evitare possibili confusioni. Editate il progetto e mediante Add Node aggiungete il file in assembler precedentemente salvato nella directory di lavoro. Selezionate File → Open e aprite il file che abbiamo creato e aggiungetelo al progetto.

Cliccate Control+F10 o utilizzate la barra degli strumenti o il menù di controllo per assemblare e compilare il codice. Il codice si compila senza errori e con due messaggi di avviso, nel caso avessimo confuso i banchi di memoria

```

Ese5 - Blocco note
File Modifica Formato Visualizza ?

;il display a 7 segmenti
;Sul display a catodo comune collegato alla porta B, visualizziamo lo stato
;logico "0" o "1" dell'interruttore RC0. Mediante l'interruttore RC1 si attiva o meno il
;punto decimale.

List      p=16F870      ;Tipo de processore
include   "P16F870.INC" ;Definizione dei registri interni

org       0x00
goto     Inizio
org       0x05

Inizio    clrwf PORTB      ;Cancella i valori casuali
          bsf STATUS,RP0   ;Seleziona il banco 1
          clrwf TRISB      ;Porta B si configura come uscita
          movlw 0xFF        ;Porta C si configura come ingresso
          movwf TRISC
          bcf STATUS,RP0    ;Seleziona il banco 0

Loop      clrwdt            ;Aggiorna il WDT
          btfscc PORTC,0    ;Verifica RC0
          goto RC0_vale_1   ;E' a livello "1"
          movlw b'00111111' ;Visualizza la cifra 0
          movwf PORTB
          goto Test_RC1

RC0_vale_1 movlw b'00000110' ;Visualizza la cifra 1
          movwf PORTB

Test_RC1  btfscc PORTC,1    ;Verifica RC1
          goto RC1_vale_1   ;E' a livello "1"
          bcf PORTB,7       ;Disattiva il punto decimale
          goto Loop

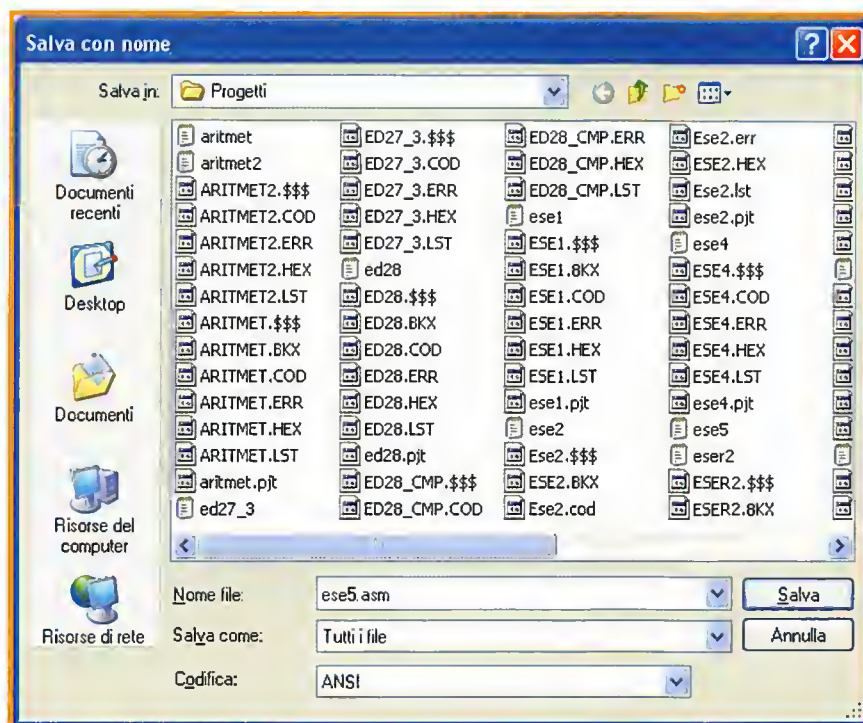
RC1_vale_1: bcf PORTB,7     ;Attiva il punto decimale
          goto Loop

end                          ;Fine del programma

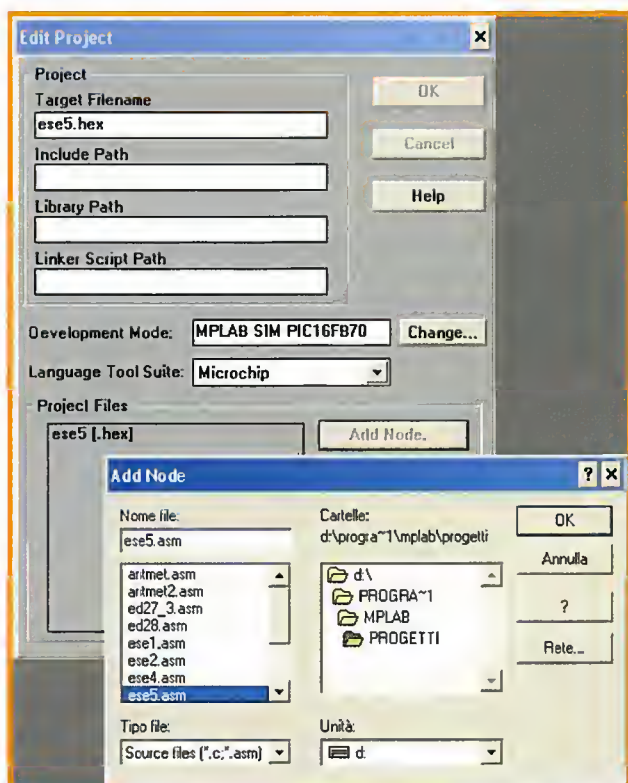
```

Codice finale del programma.





Salvate il programma  
nella directory di lavoro con l'estensione corretta.



Dobbiamo aggiungere il codice creato al progetto.

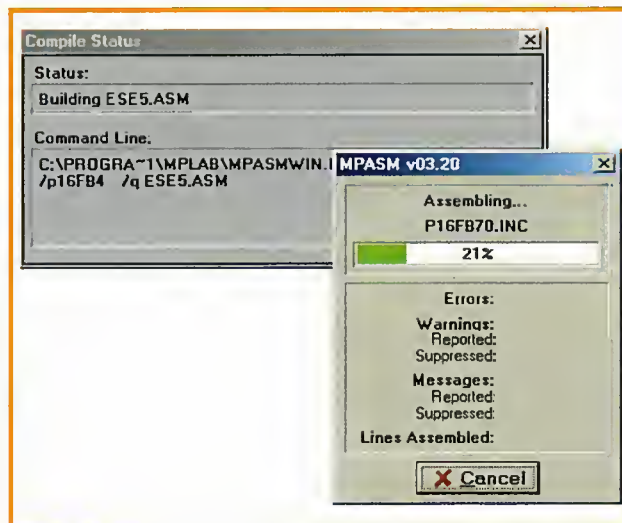
riferiti ai registri TRISx. Questi messaggi non sono importanti, ma dobbiamo comunque essere attenti se in qualche altro caso veniamo avvisati da messaggi diversi. Dobbiamo verificare sempre i messaggi forniti dal compilatore.

## Simulazione

Abbiamo eseguito ciò che per molti è la parte più complicata: la creazione del codice e la sua verifica, però come in tutti i progetti, prima di scrivere dobbiamo assicurarci che il programma esegua esattamente le nostre istruzioni. Visualizziamo le finestre dei registri delle funzioni speciali e quella che contiene i registri di ingresso e di uscita (che dovrà essere stata creata in precedenza).

Configurate quest'ultima finestra in modo che la visualizzazione si realizzi in binario. Infine, configurate il simulatore di stimoli asincroni per fare in modo che i primi due segnali siano RC0 e RC1.

Si simula direttamente, inizialmente i bit RC0 e RC1 saranno a zero, quindi sull'uscita troveremo il valore "00111111". Se ora cambiamo lo stato del pin RC1 a High sul simulatore e cliccheremo su di esso potremo osservare come cambia il valore dell'ingresso, e come



Processo di compilazione.



Visualizza i registri di  
ingresso/uscita

**Watch\_1**

Address	Symbol	Value
07	PORTC	H' 00'
06	PORTB	B' 00111111'

**Add Watch Symbol**

Symbol:

PORTB

pir2  
PIR2  
porta  
PORTA  
portb  
**PORTB**  
portc  
PORTC  
pr2  
PR2

Add  
Delete  
Close  
Properties  
Help

Uscita quando  $RC0 = RC1 = 0$ .

L'uscita prenderà il valore "10111111", ciò significa che il bit 7 di questa uscita, che corrisponde al punto decimale, è stato attivato.

Se ora attiviamo il pin  $RC0$  nello stesso modo, potremo verificare come varia il valore dell'ingresso e come viene eseguita l'altra parte del programma, con l'uscita corrispondente a 1 sul display e il punto decimale attivato.

Sul programma c'è un errore perché quando torna a ripetere il ciclo, legge solamente il pin  $RC0$  e in funzione di quest'ultimo fornisce un'uscita. L'uscita in questo caso azzerava sempre il valore di  $RB7$  (punto decimale), quindi anche se l'interruttore  $RC1$  non cambia posizione e continua a richiedere che il punto sia attivato, sino a quando il programma non tornerà a rileggerne il valore, il punto rimarrà spento.

Calcolate il tempo di durata di questo processo e cercate di correggere l'errore.

Nella nostra applicazione questo errore non ha importanza, dato che visivamente non si apprezza questo breve lasso di tempo in cui si spegne il punto decimale, ma in un'altra applicazione questo potrebbe essere un errore critico. Immaginate, per esempio, che questo inter-

ruttore controlli il pilota automatico di un aereo, e che ciclicamente venga scollegato il segnale di pilota automatico. Anche se si tratta di un caso estremo, dobbiamo vigilare su questi possibili errori e, come volevamo dimostrarvi, è nella fase di simulazione dove possiamo correggere questi problemi. Dato che per la nostra visualizzazione il funzionamento è corretto, diamo per buona questa fase e prepariamoci a realizzare il montaggio hardware dell'applicazione.

**MPLAB IDE - D:\PROGRA-1\MPLAB\PROGETTINESI.PJT**

File Project Edit Debug PICSTART Plus Options Tools Window Help

Source: d:\progra-1\mplab\progettinesi.asm

```

;Verifica RC0
;E' a livello "1"
;Visualizza la cifra 0
;Visualizza la cifra 1
;Verifica RC1
;E' a "1"
;Disattiva il punto decimale
;Attiva il punto decimale
;Fine del programma

```

**Watch\_1**

Address	Symbol	Value
07	PORTC	B' 00000000'
06	PORTB	B' 00111111'

**Special Function Register Window**

SFR Name	Hex	Dec	Binary	Char
tr0	06	6	00000110	.
tr1	08	8	00000000	.
option_reg	FF	255	11111111	.
pc1	10	24	00011000	.
pc1ath	00	0	00000000	.
Status	1C	28	00011100	.
fsr	00	0	00000000	.
porta	00	0	00000000	.
trisa	3F	63	00111111	?
portb	0A	10	00001110	.
tristb	00	0	00000000	.
portc	03	3	00000011	.
trisc	FF	255	11111111	.
intcon	00	0	00000000	.
pie1	00	0	00000000	.
pie1	00	0	00000000	.
pie2	00	0	00000000	.
pie2	00	0	00000000	.
twr11	00	0	00000000	.
twr1h	00	0	00000000	.
ticon	00	0	00000000	.
twr2	00	0	00000000	.
pr2	FF	255	11111111	.
t2con	00	0	00000000	.
ccpr1l	00	0	00000000	.
ccpr1h	00	0	00000000	.
ccpr2l	00	0	00000000	.
ccpr2h	00	0	00000000	.
ccpr3l	00	0	00000000	.
ccpr3h	00	0	00000000	.
ccpr4l	00	0	00000000	.
ccpr4h	00	0	00000000	.
ccpr5l	00	0	00000000	.
ccpr5h	00	0	00000000	.
ccpr6l	00	0	00000000	.
ccpr6h	00	0	00000000	.
ccpr7l	00	0	00000000	.
ccpr7h	00	0	00000000	.
ccpr8l	00	0	00000000	.
ccpr8h	00	0	00000000	.
ccpr9l	00	0	00000000	.
ccpr9h	00	0	00000000	.
ccpr10l	00	0	00000000	.
ccpr10h	00	0	00000000	.
ccpr11l	00	0	00000000	.
ccpr11h	00	0	00000000	.
ccpr12l	00	0	00000000	.
ccpr12h	00	0	00000000	.
ccpr13l	00	0	00000000	.
ccpr13h	00	0	00000000	.
ccpr14l	00	0	00000000	.
ccpr14h	00	0	00000000	.
ccpr15l	00	0	00000000	.
ccpr15h	00	0	00000000	.
ccpr16l	00	0	00000000	.
ccpr16h	00	0	00000000	.
ccpr17l	00	0	00000000	.
ccpr17h	00	0	00000000	.
ccpr18l	00	0	00000000	.
ccpr18h	00	0	00000000	.
ccpr19l	00	0	00000000	.
ccpr19h	00	0	00000000	.
ccpr20l	00	0	00000000	.
ccpr20h	00	0	00000000	.
ccpr21l	00	0	00000000	.
ccpr21h	00	0	00000000	.
ccpr22l	00	0	00000000	.
ccpr22h	00	0	00000000	.
ccpr23l	00	0	00000000	.
ccpr23h	00	0	00000000	.
ccpr24l	00	0	00000000	.
ccpr24h	00	0	00000000	.
ccpr25l	00	0	00000000	.
ccpr25h	00	0	00000000	.
ccpr26l	00	0	00000000	.
ccpr26h	00	0	00000000	.
ccpr27l	00	0	00000000	.
ccpr27h	00	0	00000000	.
ccpr28l	00	0	00000000	.
ccpr28h	00	0	00000000	.
ccpr29l	00	0	00000000	.
ccpr29h	00	0	00000000	.
ccpr30l	00	0	00000000	.
ccpr30h	00	0	00000000	.
ccpr31l	00	0	00000000	.
ccpr31h	00	0	00000000	.
ccpr32l	00	0	00000000	.
ccpr32h	00	0	00000000	.
ccpr33l	00	0	00000000	.
ccpr33h	00	0	00000000	.
ccpr34l	00	0	00000000	.
ccpr34h	00	0	00000000	.
ccpr35l	00	0	00000000	.
ccpr35h	00	0	00000000	.
ccpr36l	00	0	00000000	.
ccpr36h	00	0	00000000	.
ccpr37l	00	0	00000000	.
ccpr37h	00	0	00000000	.
ccpr38l	00	0	00000000	.
ccpr38h	00	0	00000000	.
ccpr39l	00	0	00000000	.
ccpr39h	00	0	00000000	.
ccpr40l	00	0	00000000	.
ccpr40h	00	0	00000000	.
ccpr41l	00	0	00000000	.
ccpr41h	00	0	00000000	.
ccpr42l	00	0	00000000	.
ccpr42h	00	0	00000000	.
ccpr43l	00	0	00000000	.
ccpr43h	00	0	00000000	.
ccpr44l	00	0	00000000	.
ccpr44h	00	0	00000000	.
ccpr45l	00	0	00000000	.
ccpr45h	00	0	00000000	.
ccpr46l	00	0	00000000	.
ccpr46h	00	0	00000000	.
ccpr47l	00	0	00000000	.
ccpr47h	00	0	00000000	.
ccpr48l	00	0	00000000	.
ccpr48h	00	0	00000000	.
ccpr49l	00	0	00000000	.
ccpr49h	00	0	00000000	.
ccpr50l	00	0	00000000	.
ccpr50h	00	0	00000000	.
ccpr51l	00	0	00000000	.
ccpr51h	00	0	00000000	.
ccpr52l	00	0	00000000	.
ccpr52h	00	0	00000000	.
ccpr53l	00	0	00000000	.
ccpr53h	00	0	00000000	.
ccpr54l	00	0	00000000	.
ccpr54h	00	0	00000000	.
ccpr55l	00	0	00000000	.
ccpr55h	00	0	00000000	.
ccpr56l	00	0	00000000	.
ccpr56h	00	0	00000000	.
ccpr57l	00	0	00000000	.
ccpr57h	00	0	00000000	.
ccpr58l	00	0	00000000	.
ccpr58h	00	0	00000000	.
ccpr59l	00	0	00000000	.
ccpr59h	00	0	00000000	.
ccpr60l	00	0	00000000	.
ccpr60h	00	0	00000000	.
ccpr61l	00	0	00000000	.
ccpr61h	00	0	00000000	.
ccpr62l	00	0	00000000	.
ccpr62h	00	0	00000000	.
ccpr63l	00	0	00000000	.
ccpr63h	00	0	00000000	.
ccpr64l	00	0	00000000	.
ccpr64h	00	0	00000000	.
ccpr65l	00	0	00000000	.
ccpr65h	00	0	00000000	.
ccpr66l	00	0	00000000	.
ccpr66h	00	0	00000000	.
ccpr67l	00	0	00000000	.
ccpr67h	00	0	00000000	.
ccpr68l	00	0	00000000	.
ccpr68h	00	0	00000000	.
ccpr69l	00	0	00000000	.
ccpr69h	00	0	00000000	.
ccpr70l	00	0	00000000	.
ccpr70h	00	0	00000000	.
ccpr71l	00	0	00000000	.
ccpr71h	00	0	00000000	.
ccpr72l	00	0	00000000	.
ccpr72h	00	0	00000000	.
ccpr73l	00	0	00000000	.
ccpr73h	00	0	00000000	.
ccpr74l	00	0	00000000	.
ccpr74h	00	0	00000000	.
ccpr75l	00	0	00000000	.
ccpr75h	00	0	00000000	.
ccpr76l	00	0	00000000	.
ccpr76h	00	0	00000000	.
ccpr77l	00	0	00000000	.
ccpr77h	00	0	00000000	.
ccpr78l	00	0	00000000	.
ccpr78h	00	0	00000000	.
ccpr79l	00	0	00000000	.
ccpr79h	00	0	00000000	.
ccpr80l	00	0	00000000	.
ccpr80h	00	0	00000000	.
ccpr81l	00	0	00000000	.
ccpr81h	00	0	00000000	.
ccpr82l	00	0	00000000	.
ccpr82h	00	0	00000000	.
ccpr83l	00	0	00000000	.
ccpr83h	00	0	00000000	.
ccpr84l	00	0	00000000	.
ccpr84h	00	0	00000000	.
ccpr85l	00	0	00000000	.
ccpr85h	00	0	00000000	.
ccpr86l	00	0	00000000	.
ccpr86h	00	0	00000000	.
ccpr87l	00	0	00000000	.
ccpr87h	00	0	00000000	.
ccpr88l	00	0	00000000	.
ccpr88h	00	0	00000000	.
ccpr89l	00	0	00000000	.
ccpr89h	00	0	00000000	.
ccpr90l	00	0	00000000	.
ccpr90h	00	0	00000000	.
ccpr91l	00	0	00000000	.
ccpr91h	00	0	00000000	.
ccpr92l	00	0	00000000	.
ccpr92h	00	0	00000000	.
ccpr93l	00	0	00000000	.
ccpr93h	00	0	00000000	.
ccpr94l	00	0	00000000	.
ccpr94h	00	0	00000000	.
ccpr95l	00	0	00000000	.
ccpr95h	00	0	00000000	.
ccpr96l	00	0	00000000	.
ccpr96h	00	0	00000000	.
ccpr97l	00	0	00000000	.
ccpr97h	00	0	00000000	.
ccpr98l	00	0	00000000	.
ccpr98h	00	0	00000000	.
ccpr99l	00	0	00000000	.
ccpr99h	00	0	00000000	.
ccpr100l	00	0	00000000	.
ccpr100h	00	0	00000000	.
ccpr101l	00	0	00000000	.
ccpr101h	00	0	00000000	.
ccpr102l	00	0	00000000	.
ccpr102h	00	0	00000000	.
ccpr103l	00	0	00000000	.
ccpr103h	00	0	00000000	.
ccpr104l	00	0	00000000	.
ccpr104h	00	0	00000000	.
ccpr105l	00	0	00000000	.
ccpr105h	00	0	00000000	.
ccpr106l	00	0	00000000	.
ccpr106h	00	0	00000000	.
ccpr107l	00	0	00000000	.
ccpr107h	00	0	00000000	.
ccpr108l	00	0	00000000	.
ccpr108h	00	0	00000000	.
ccpr109l	00	0	00000000	.
ccpr109h	00			